



Veliev, Erdoan and Penev, Kalin. (2008). Visualization of Free Search Process. In: Advanced Topics on Evolutionary Computing. WSEAS Press, Bulgaria, pp. 133-135. ISBN 978 960 6766 58 9

Downloaded from <http://ssudl.solent.ac.uk/269/>

Usage Guidelines

Please refer to usage guidelines at <http://ssudl.solent.ac.uk/policies.html> or alternatively contact ir.admin@solent.ac.uk.

Visualization of Free Search Process

Erdoan Veliev

Faculty of Computing Systems and Technologies
Technical University of Sofia
1000 Sofia Technical University str. Kliment Ohridski 8,
BULGARIA
erdoan.veliev@gmail.com <http://www.tu-sofia.bg>

Kalin Penev

Southampton Solent University
East Park Terrace, Southampton,
Hampshire SO14 0YN, UK
Kalin.Penev@solent.as.uk www.solent.ac.uk

Abstract: - The article presents visualization of an adaptive method for optimization called Free Search and gives a description of methodology used. Implemented tasks are illustrated with relevant graphics. Benefits of having 3D graphical interface are in opportunity to observe and track the changes and behavior of this algorithm during the search process. Technologies and tools used for visualization are explained and discussed.

Key-Words: - Free Search Algorithm, Test Functions, Visualization, Java Programming Language, Java 3D

1 Introduction

The paper presents how is visualized an advance method for search and optimization – Free Search[2][3][4].

Free Search is an adaptive method for search and optimization. The method models the behavior of the animals in the natural world. The individuals of the population are called animals. The animals in the model have individual sense, which is ability for orientation in the search space. The animals follow their own sense. They know for the results of previous explorations, it is taken under attention, but this not make them to go into explicit direction. They can select any other search direction. In the algorithm the mobility and the sense are related, it is fundamental for the algorithm [3].

Using 3D graphical application will provide opportunity to analyze and evaluate the received result from different test functions when Free Search Algorithm is applied to them. The test functions could be Himmelblau function, Keane function, Three Hills function and etc.[3]

2 Visualization Problem

There are a number of search algorithms giving different results. Free Search, Differential Evolution, Genetic algorithms – all they are methods for search and optimization.

The purpose of the project is to provide a 3D graphical interface. Visualizing the search process for different advanced search algorithms will help to track the time dependent changes during search process. It is

simpler to explore the graphic's change when it is presented in 3D graphical mode. For example it is easier to trace the peak of search process. By giving various color schemes of the different parts of the graph, analyzing the achieved results becomes more convenient.

2.2 Visualization of test function graph

Visualization of test function's graph presents the shape of explored test function in three dimensional space. By rotating and scaling the graph, global minimums and maximums are easily explored and estimated.

2.2 Visualization of the search process

Visualization of the search process actually is displaying the steps of the individuals after each iteration. The aim is to be shown the exact way that every animal covers while searching the best possible solution.

2.2.1 Static visualization

Visualization of the steps of the individuals when the search space is constant in the time. The optimal minimums and maximums don't leave the searching area.

2.2.1 Dynamic visualization

The search space is variable in time, depending on the iterations that individuals have made. In given moment optimal maximum or maximums leave the explored area.

3 Problem Solution

There are many technologies, programming languages and tools for implementing such 3D graphical application. The goal is to choose the most appropriate solution for reaching the final aim.

3.1 Platform Selection

There are a number of platforms, which can be used for resolving the aim of the project – development 3D graphic interface – Java 3D API, MFC, OpenGL.

Java platform independence is a good reason for choosing it. Java 3D API [8] allows programming in 100 percent java and running everywhere. “It is therefore very attractive to have all the application code, persistence, and user interface code in an easily portable language, such as Java” [1].

Combining the Java Applet Technology [9] [10] and Java 3D API allows development a web based application which could be used by wide range of people all over the world.

3.2 Database Selection

Displaying the individuals on the 3D graphical scene and the movements they are making, requires coordinates of each animal for each iteration. This data is computed during the search process, which is separated from the visualization process. There are two appropriate possibilities that could be chosen to store the data. The first is using database servers like Oracle, MySQL, Informix and etc. The second opportunity is using the local file system. Due to the small size of the objects which have to be stored and the additional efforts for maintaining the database servers, there is no need of using them. According to this, the file system option is more acceptable for solution the problem.

3.3 Visualization of the test function graph

For three dimensional representation of the test function graph is used Java 3D conceptual model for image rendering. The shape of the function graph is created by calculation the z coordinate, which is test function value for each X and Y from the function area.

```
for (x = xMin; x <= xMax; x += step) {
    for (y = yMin; y <= yMax; y += step) {
        s1 = x * x + y - 11;
        s2 = x + y * y - 7;
        f = 200 - s1 * s1 - s2 * s2;
        points[xLine][yLine] = new Point3d(x, y, z);
        yLine++;
    }
    xLine++;
}
```

Fig.1. Himmelblau test function

Analytic representation of Himmelblau test function:

$$F = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2 \quad [5]$$

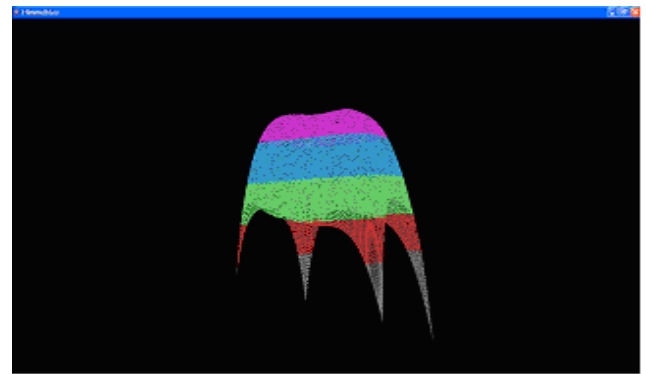


Fig.2. Visual representation of Himmelblau test function in three dimensional space.

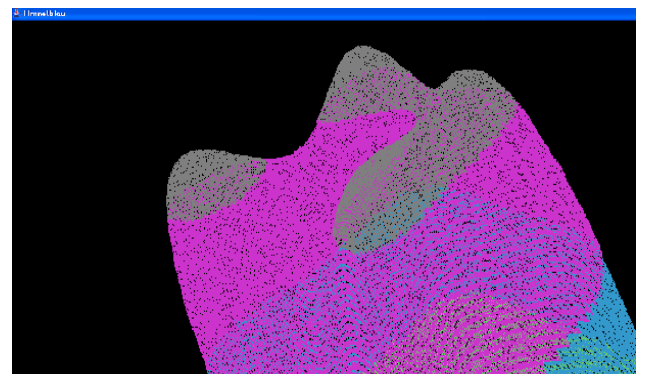


Fig 3. The four equal maximums of Himmelblau func.

3.4 Visualization of the search process

Displaying the steps of the animals during search process is made by some delay between the iterations to clarify the direction they are moving.

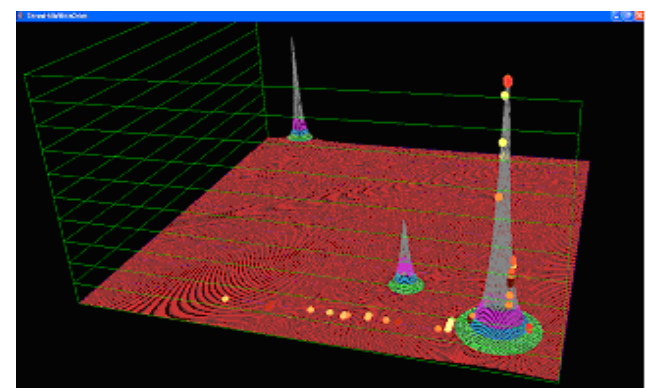


Fig. 4. Three Hills test function

The spheres on the scene in Fig 4 are the animals in the search process. Every animal is displayed with different

color to be easy traceable. It is clearly shown that some of the individuals find the maximum and others can't.

Three Hills Function:

$$F = \sum_{j=1}^m \{z_j / [1 + k_j \sum_{i=1}^n (a_{ji} + dA_{ji}(t) - x_i)^2]\}$$

where x_i are the variables, m is number of peaks, $j = 1..m$, n is number of dimensions $i=1..n$, z_j are the heights of the peaks, k_j are the widths of the peaks, a_{ji} are the initial locations of the peaks, $dA_{ji}(t)$ are the time dependent components, t is the time dependent component, which defines the period for change.

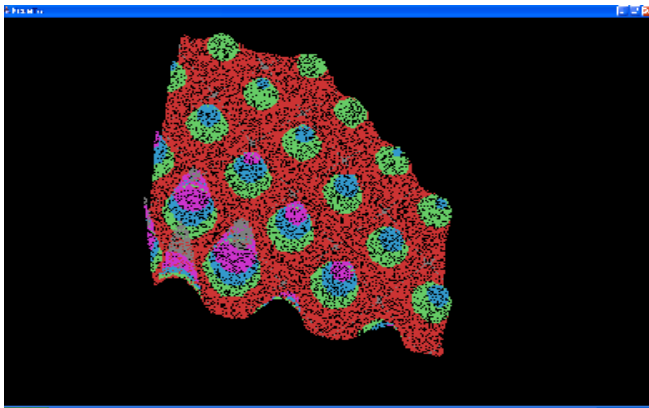


Fig. 5. Keane test function – constraint area is visible.

Keane Test Function:

$$f(x_i) = \left| \sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i) \right| / \sqrt{\sum_{i=1}^n x_i^2}$$

for : $0 < x_i < 10$ and $i=1..n$ subject to $\prod_{i=1}^n x_i > 0.75$ and

$\sum_{i=1}^n x_i < 15n/2$ starting from $x_i = 5$, $i=1..n$ where x_i are the

variables expressed in radians and n is the number of dimensions. [6]

4 Conclusion

Achieving good results in analyzing of so complicated algorithm such as Free Search is impossible without visual representation of the search process. "The behaviour of the animals in FS is not defined. It is not strictly controlled and can be described by some probability, only. They have a freedom to make a

decision where and how to search." [2] According to this, there is a good reason to develop 3D graphical software to visualize the process. Implemented methodology could contribute significantly for observation and analysis of the optimisation. Further research could focus on visualisation of other tests functions, time dependent problems and real world tasks.

References:

- [1] Selman Daniel – Java 3D Programming
- [2] Penev Kalin - Free Search in Tracking Time Time Dependent Optima
- [3] Penev Kalin – Free Search – Extensive Evaluation 2006 Parmee I.C. editor, *Adaptive Computing in Design and Manufacturing VII* pp 157-165
- [4] Penev Kalin - Adaptive Computing in Support of Traffic Management 2004 Parmee I.C. editor, *Adaptive Computing in Design and Manufacturing V* pp 295-306
- [5] Himmelblau D., 1972, *Applied Non-linear Programming*, McGraw-Hill, New York, 1972.
- [6] Keane A. J., 1996, A Brief Comparison of Some Evolutionary Optimization Methods, In V.Rayward-Smith, I. Osman, C. Reeves and G.D. Smith, J. Wiley (Editors), *Modern Heuristic Search Methods*.
- [7] Parmee I.C., 2004, *Adaptive computing in design and manufacture VI*, Springer - Verlag London Limited.
- [8] API Specifications, last visited at 04.03.2008 <http://java.sun.com/products/java-media/3D/>
- [9] 2000 Michael McGuffin, Java Applet Tutorial, last visited at 04.03.2008 <http://www.dgp.toronto.edu/~mjmcguff/learn/java/>
- [10] Java Applet Tutorial, last visited at 04.03.2008 <http://www.realapplets.com/tutorial/chapter1.html>